

**19 BUNDESREPUBLIK
DEUTSCHLAND**



**DEUTSCHES
PATENTAMT**

⑫ **Offenlegungsschrift**
⑩ **DE 43 16 813 A 1**

**(51) Int. Cl.⁵:
H 03 M 13/00**

(21) Aktenzeichen: P 43 16 813.2
 (22) Anmeldetag: 19. 5. 93
 (43) Offenlegungstag: 24. 11. 94

DE 43 16 813 A 1

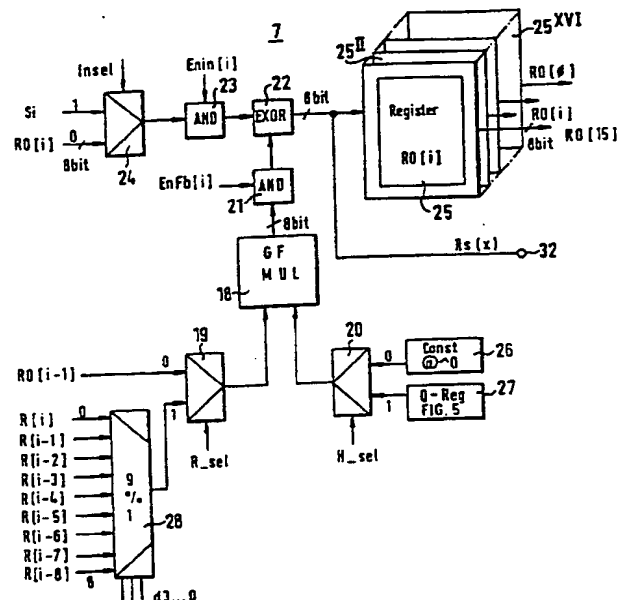
71) Anmelder:
Philips Patentverwaltung GmbH, 20097 Hamburg, DE

(72) Erfinder:
Mester, Roland, 6100 Darmstadt, DE

⑤④ Reed-Solomon-Decoder

57) Es wird ein Reed-Solomon-Decoder vorgeschlagen, der eine Einrichtung (6) zur Erzeugung von Fehlerwert- und Fehlerort-Polynomen anhand ermittelter Syndrome $S(x)$ und Auslöschorte $L(x)$ nach einem Euclid-Algorithmus aufweist. Dabei wird zur Erzeugung des Fehlerwertpolynoms $R_s(x) = (Q_{s-1}(x))R_{s-1}(x) + R_{s-2}(x)$ ein erster Galois-Feld-(GF)-Multiplizierer (18) verwendet, dem einseitig von einem ersten Register (27) ein Koeffizient des Zwischenwertpolynoms $Q_{s-1}(x) = R_{s-2}(x)/R_{s-1}(x)$ sowie ein von einem zweiten Registerstapel (61) abgreifbaren Fehlerwertpolynoms $R_{s-1}(x)$ zugeführt ist. Ein Ausgangssignal des GF-Multiplizierers (18) wird in einer EXOR-Stufe (22) dem am Ausgang eines dritten Registerstapels (58) vorliegenden Fehlerwertpolynom $R_{s-2}(x)$ additiv zugesetzt. Für die Erzeugung eines Fehlerortpolynoms $T_s(x) = (Q_{s-1}(x))T_{s-1}(x) + T_{s-2}(x)$ ist ein weiterer GF-Multiplizierer (52) vorgesehen, dem ebenfalls das Zwischenwertpolynom $Q_{s-1}(x)$ sowie ein am Ausgang eines vierten Registerstapels (61) abnehmbaren Fehlerortpolynoms $T_{s-1}(x)$ aufgeschaltet ist.

Ausgangssignale des GF-Multiplizierers (52) sind in einem EXOR-Gatter (56) einem am Ausgang eines fünften Registerstapels (58) vorliegenden Fehlerortpolynom $T_{s-2}(x)$ zuaddiert. Die Koeffizienten des Zwischenwertpolynoms $Q_{s-1}(x)$ erzeugt eine sequentielle Einrichtung (33 bis 40) zur Division der Ausgangsdaten des dritten und vierten Registerstapels (58, 61).



DE 43 16 813 A 1

Beschreibung

Die Erfindung geht aus von einem Reed-Solomon-(RS)-Decoder nach dem Oberbegriff des Patentanspruchs

1.

Ein RS-Code ist ein systematischer Blockcode der Länge n , bei welchem k Informationssymbolen ($n - k = p$ Checkworte) zugesetzt werden. Durch den Codiervorgang werden Nachrichtensymbole nicht verändert. Der RS-Code hat eine minimale Hamming-Distanz $d = p + 1$, die bei einer Decodierung frei wählbar zur Fehlererkennung oder Fehlerkorrektur eingesetzt werden kann.

Bei einer Decodierung wird nicht zwischen Daten- und Checkworten unterschieden. Ein RS-Code-Block besteht aus n Symbolen, wobei für ein Symbol jede Bit-Anzahl > 1 möglich ist. Üblicherweise wird mit einer Symbolbreite von 8 Bit gearbeitet. Für jeden Symbolfehler t werden bei einer Korrektur $2t$ Checkworte benötigt, um einen Fehlerort X_k und einen Fehlerwert Y_k zu berechnen. Ist das fehlerhafte Symbol als Auslöschung markiert, ist der Fehlerort X_k bekannt. In diesem Fall wird zur Berechnung des Fehlerwertes Y_k nur ein einziges Checkwort benötigt. Für RS-Codes gilt die Beziehung $2t + e \leq p = n - k = d - 1$, wobei mit e die Anzahl der Auslöschungen bezeichnet ist. In Zusammenhang mit der Aufzeichnung und Wiedergabe von Datensignalen auf/von Magnetband sind unter dem Begriff Auslöschungen Drop-Out-Störungen im wiedergegebenen Datensignal oder auch der kurzzeitige Ausfall eines Übertragungskanal und ein damit begründeter Datenausfall bei der digitalen Übertragung zu verstehen.

Definitionsgemäß ist bei RS-Codes jedes Symbol mit m -Bits das Element eines Galois-Feldes $GF(2^m)$. Die Sequenz der $2^m - 1$ Symbole ungleich Null als Basis der finiten Feld-Arithmetik wird durch das Feldgeneratorpolynom $p(X)$ definiert. Die Anzahl der Symbole beschränkt auch die maximale Blocklänge eines RS-Codes auf $2^m - 1$, da die Elemente des Feldes als eindeutige Positionsanzahl im Block benötigt werden.

Ein Feld wird über ein irreduzibles Feldgeneratorpolynom definiert. Wird ein Symbol als primitives Element α der Ordnung $2^m - 1$ definiert, lassen sich alle anderen Elemente als Potenz von α ausdrücken. Das Galois-Feld läßt sich somit als endliche Folge von $2^m - 1$ Elementen definieren. Dabei ermöglicht die binäre Komponenten-Darstellung eine einfache Addition über eine bitweise EXOR-Verknüpfung und eine Exponentendarstellung eine einfache Multiplikation über eine Addition der Exponenten modulo $q = 2^m - 1$. Dies entspricht in einer Schaltungsumsetzung einem binären Addierer, dessen Übertragsausgang mit dem Übertragsingang verbunden ist.

Prinzipiell ist ein RS-Code der Distanz d über das Code-Generator-Polynom

$$G(x) = \prod_{i=b}^{b+(d-2)} (x + \alpha^i)$$

definiert. Wesentlich ist, daß $G(X)$ p aufeinanderfolgende Nullstellen enthält.

Ein legaler Codevektor $c(X)$ entsteht durch Division eines um p Stellen verschobenen Informations-Vektors $i(X) \cdot x^p$ durch $G(X)$ und Anhängen des dabei entstehenden Paritäts-Vektors $p(X)$ in den freigewordenen Stellen des Informations-Vektors $i(X)$. Der resultierende Code-Vektor $c(X)$ ist dabei durch alle Wurzeln von $G(X)$ ohne Rest teilbar. Die einzelnen Polynome und Vektoren lassen sich wie folgt darstellen:

$i(x) = i_{k-1}x^{k-1} + i_{k-2}x^{k-2} + \dots + i_1x^1 + i_0$	Informations-Polynom
$p(x) = (i(x) \cdot x^p) \bmod G(x)$	Paritäts-Polynom
$c(x) = (i(x) \cdot x^p) + p(x)$	Code-Polynom
$c_i = (i_{k-1}, i_{k-2}, \dots, i_1, i_0, p_{p-1}, p_{p-2}, \dots, p_1, p_0) = (c_{n-1}, c_{n-2}, \dots, c_1, c_0)$	Code-Vektor = gesendeter Vektor
$e_i = (e_{n-1}, e_{n-2}, \dots, e_1, e_0)$	Fehler-Vektor
$u_i = (u_{n-1}, u_{n-2}, \dots, u_1, u_0)$	Auslöschungs-Vektor
$r_i = c_i + e_i + u_i$	empfangener Vektor

Der Fehler-Vektor e_i ist an maximal t Stellen ungleich Null, ansonsten gleich Null. Entsprechendes gilt für den Auslöschungs-Vektor u_i , der an maximal e Stellen ungleich Null ist. Bei Überschneidungen von e_i und u_i an gleichen Positionen werden Fehler zu Auslöschungen, da die Fehler dann markiert sind.

Ein übertragener RS-codierter Codeblock $c(X)$ besteht aus k Informationsstellen, denen p Checkworte folgen. Durch die Codekonstruktion ist gewährleistet, daß $c(X)$ durch alle Wurzeln durch $G(X)$ ohne Rest teilbar ist, solange keine Fehler $e(X)$ oder Auslöschungen $u(X)$ auftreten. Die Entwicklung eines Empfangsvektors $r(X)$ an den Wurzeln des Code-Generator-Polynoms $G(X)$ erzeugt Werte, die von empfangenen Fehlern abhängig sind, jedoch unabhängig von der übertragenen Information $i(X)$. Diese Werte werden als Syndrome bezeichnet. Die m Syndrome S_j ($j = 0 \dots d-2$) werden nach der Gleichung

$$S_{b+j} = \sum_{i=0}^{n-1} r_i (\alpha^{i \cdot (b+j)}) = \sum_{k=0}^{e+t-1} Y_k (X_k)^{(b+j)}$$

berechnet.

Der empfangene Vektor (x) läßt sich in einer Polynom-Darstellung wie folgt beschreiben:

$$r(x) = r_{n-1}x^{n-1} + r_{n-2}x^{n-2} + \dots + r_1x^1 + r_0$$

Dabei bezeichnet x^i eine Position im Polynom für jedes empfangene Symbol r_i . Das erste empfangene (Daten-) Symbol ist das letzte (Paritäts-) Symbol r_0 . Zur Speicherung des Polynoms müssen nur die Koeffizienten r_i abgelegt werden.

Da die Entwicklung eines fehlerfreien Empfangsvektors $r(X)$ an einer Nullstelle des Codegenerator-Polynoms den Wert Null liefert, sind die Werte der Syndrome als Summen der positionsgewichteten Fehler Y_k an den Stellen X_k zu betrachten. Die Syndrome stellen dabei Reste einer Division des Empfangsvektors $r(X)$ durch alle Faktoren $(x + \alpha^i)$ des Codegeneratorpolynoms dar, wobei

$$S_j = \sum_{k=0}^{e+t-1} Y_k (X_k)^j \quad (\text{für alle Syndrome } S_0 \dots S_{d-2} \text{ im Fall } b=0)$$

ist.

Wird bei einer Auslöschung die Position eines Fehlers X_k festgehalten, so läßt sich mit p Syndromen und p Fehlerorten X_k bis zu $e = p$ Auslöschungen berechnen, so lange $2t + e \leq p$ ist, wobei t die Anzahl nicht markierten Fehler ist.

Die Syndrome bilden nach der zuvor genannten Gleichung einen Satz von p nichtlinearen Gleichungen, so daß p Lösungen ermittelt werden können. Gesucht werden mit Hilfe der Syndrome die Koeffizienten eines Fehlerortpolynoms, welches an den Fehlerorten den Wert Null aufweist.

Eine Decodierung von RS-codierten Datenblöcken nach dem sogenannten Euclid-Algorithmus ist von Truong, Eastman, Reed und Hsu in der Zeitschrift IEE Proceedings, "Simplified procedure for correcting both errors and erasures of RS code using Euclid algorithm", Vol. 135, Pt.E, No. 6, Nov. 1988 angegeben. Der Euclid-Algorithmus läßt sich in folgende Schritte gliedern:

1. Berechnung der Syndrome S_j sowie des Polynoms von Auslöschungsorten $L(X)$ und dem Setzen von $e = \deg(L(X))$ (=Anzahl der Auslöschungen).
2. Berechnung eines sogenannten Forney-Syndroms $\tau(X) = S(X)L(X) \bmod x^{d-1}$.
3. Für $e = d-1 = p$ wird $T(X) = L(X)$ und $R(X) = \tau(X)$ gesetzt. Anderenfalls wird $T(X)$ und $R(X)$ durch Anwendung eines Euclid-Algorithmus bestimmt. Der Euclid-Algorithmus besteht aus folgenden rekursiven Formeln:

$$\begin{aligned} Q_{s-1}(X) &= R_{s-2}(X)/R_{s-1}(X), \\ T_s(X) &= (Q_{s-1}(X))T_{s-1}(X) + T_{s-2}(X) \\ R_s(X) &= (Q_{s-1}(X))R_{s-1}(X) + R_{s-2}(X) \end{aligned}$$

mit den Anfangswerten:

$T_0(X) = L(X)$, $T_{-1}(X) = 0$, $R_{-1}(X) = x^{d-1}$, $R_0(X) = \tau(X)$. Darin ist $T_s(X)$ ein Fehlerortpolynom, $R_s(X)$ ein Fehlerwertpolynom und $Q_{s-1}(X)$ ein Zwischenwertpolynom. Danach kann $R_s(X)$ und $T_s(X)$ mit dem niedrigsten Koeffizienten $T_s(0) = \sigma$ normiert werden, wobei $R(X) = R_s(X)/\sigma$ ist. $R(X)$ und $T(X)$ sind die Ergebnisse des Euclid-Algorithmus, wenn der Grad $R_s \leq \lfloor (d + e - 3)/2 \rfloor$ ist, wobei $\lfloor \cdot \rfloor$ einen ganzzahligen Anteil bezeichnen.

Aus dem Tagungsband IEEE 1990 Custom Integrated Circuits Conference, "A 40 Mhz Encoder/Decoder Chip generated by a Reed-Solomon Code Compiler", Seiten 13.5.1 bis 13.5.4, CH2860-5/90/0000-0065, ist bereits eine Schaltungsanordnung zum Decodieren von RS-codierten Daten bekannt.

Bei dieser bekannten Schaltungsanordnung werden zunächst in einer ersten Stufe von den empfangenen Datenworten einer Übertragungsstrecke Syndrome berechnet. In einer nachgeschalteten zweiten Stufe werden von den berechneten Syndromen nach dem Euclid-Algorithmus Fehlerort- und Fehlerwert-Polynome berechnet. Die bekannte Schaltungsanordnung weist jedoch den Nachteil auf, daß Auslöschungen weder berücksichtigt und noch korrigiert werden können.

Weiterhin ist aus der DE-Zeitschrift Elektronik, 25/1992, Seiten 46 bis 52, sowie 26/1992, Seiten 40 bis 44, insbesondere Bild 6, eine Anordnung zur Berechnung von $T_s(X)$ und $R_s(X)$ nach dem Euclid-Algorithmus bekannt, welche vier Registerbänke (R , RO , T und TO) mit 16 und 17 Byte Breite umfaßt. Die Registerbänke können aus verschiedenen Funktionsblöcken geladen werden. In die Registerbänke R und RO werden die ermittelten Syndrome S und Auslöschorte L geladen, während die Registerbänke T und TO konstante Anfangswerte 0 bzw. 1 erhalten. Die Registerbänke R und T können untereinander vertauscht werden, so daß Werte auch zwischengespeichert werden können. Eine "Shift RO"-Funktion wird für die Berechnung des zuvor erwähnten Orney-Syndroms benötigt. Die Registerbänke RO und TO können mit neu berechneten Polynomen zur Ausführung des Euclid-Algorithmus geladen werden. Der Grad von T zeigt am Ende der Berechnung an, wieviel Fehler in einem Datenblock für eine nachfolgende Suche nach Chien erwartet werden. Eine zentrale Steuerung erkennt dabei das Terminieren des Algorithmus über den Grad von RO .

Der vorliegenden Erfindung liegt die Aufgabe zugrunde, einen RS-Decoder nach der eingangs genannten Art zu schaffen, welcher echtzeitfähig mit hoher Datenrate (≤ 100 Mbit/s) aus den vorliegenden Syndromen und Auslöschorten des zu decodierenden Datensignals gemäß einem Euclid-Algorithmus die Fehlerort- und Fehlerwert-Polynome ermittelt.

5 Diese Aufgabe wird durch die im kennzeichnenden Teil des Patentanspruchs 1 angegebenen Merkmale gelöst. Die Erfindung weist den Vorteil auf, daß die Einrichtung zur Erzeugung der Fehlerort- und Fehlerwert-Polynome nicht nur in sehr wenigen Taktschritten die zur weiteren Decodierung erforderlichen Polynome zur Verfügung stellt, sondern daß die Einrichtung auch in Abhängigkeit berechneter Zwischenergebnisse in ihrer Funktion beeinflußt werden kann.

10 Durch die in den Unteransprüchen aufgeführten Maßnahmen sind vorteilhafte Weiterbildungen und Verbesserungen des im Patentanspruch 1 angegebenen RS-Decoders möglich.

Ein Ausführungsbeispiel der Erfindung ist in der Zeichnung dargestellt und in der nachfolgenden Beschreibung näher erläutert. Es zeigen:

Fig. 1 das grobe Blockschaltbild eines bekannten RS-Decoders,

15 Fig. 2 das Blockschaltbild einer bekannten Einrichtung zur Erzeugung von Fehlerwert- und Fehlerort-Polynomen,

Fig. 3 das Blockschaltbild einer Registertransfereinrichtung zur Erzeugung des Fehlerwertpolynoms $R_s(X)$ gemäß der Erfindung,

20 Fig. 4 das Blockschaltbild einer Registertransfereinrichtung zur Erzeugung des Fehlerwert-Polynoms $R_s(X)$ gemäß der Erfindung,

Fig. 5 das Blockschaltbild einer Steuereinrichtung für die in den Fig. 3, 4, 6 und 7 dargestellten Registertransfereinrichtungen,

Fig. 6 das Blockschaltbild einer Registertransfereinrichtung zur Erzeugung des Fehlerort-Polynoms $T_s(X)$ gemäß der Erfindung,

25 Fig. 7 das Blockschaltbild einer Registertransfereinrichtung zur Erzeugung des Fehlerort-Polynoms $T_{s-1}(X)$ gemäß der Erfindung und

Fig. 8 bis 11 Wahrheitstabellen für die Steuerung der in den Fig. 3, 4, 6 und 7 dargestellten Registertransfereinrichtungen.

30 Die Fig. 1 zeigt das grobe Blockschaltbild eines RS-Decoders. Darin ist mit 1 ein Datenbus bezeichnet, über welchem 8 Bit breite Datenworte D_{in} der zu decodierenden Datenblöcke zu einer Verzögerungseinrichtung 3 sowie einen Funktionsblock 3 übertragen werden. Der Funktionsblock 3 dient zur Ermittlung von Syndromen $S(X)$ und Auslöschorten $L(X)$. Parallel zu den auf dem Datenbus 1 übertragenen Datenblöcken wird ein Fehlersignal E_{Fin} über eine Leitung 4 der Verzögerungsleitung 2 und dem Funktionsblock 3 zugeführt.

35 Der Funktionsblock 3 enthält 16 Syndromgeneratoren zur Berechnung der Syndrome $S(X)$. Die Syndromgeneratoren bilden den Rest einer Division $c(X)/(x + \alpha^i)$. Neben den Syndromen $S(X)$ werden in dem Funktionsblock 3 die Auslöschorte $L(X)$ ermittelt, indem ein Generator alle Positionen im Datenblock durchzählt. Bei dem Auftreten einer Auslöschung wird der Ort in einem Registerstapel gespeichert. Treten mehr Auslöschungen auf, als aufgrund einer Begrenzung der Korrektur möglich sind, wird $L(X)$ Null gesetzt. In diesem Fall kann eine begrenzte Korrektur von Fehlern versucht werden. Weiterhin weist der Funktionsblock 3 einen Eingang 5 auf, 40 über welchem die Anzahl von Parity-Worten P pro Datenblock angewählt werden kann. Ausgänge des Funktionsblockes 3 sind über Leitungen P , $S00 \dots S15$ und $L00 \dots L15$ mit Eingängen eines nachgeschalteten Funktionsblockes 6 verbunden. Die Leitung P dient zur Übertragung der Anzahl der verwendeten Parity-Worte P , die Datenbusleitung $S00 \dots S15$ zur Weiterleitung von 16×8 Bit breiten Syndromvektoren $S(X)$ und die Datenbusleitung $L00 \dots L15$ zur Weiterleitung von 16×8 Bit breiten Auslöschungsvektoren $L(X)$.

45 Der in der Fig. 2 ausführlicher dargestellte Block 6 umfaßt vier Registerblöcke 7, 8, 9 und 10 mit 16 und 17 Byte Breite. Ferner umfaßt der Funktionsblock 6 eine Steuereinrichtung 11, in welcher das Zwischenwert-Polynom $Q_{s-1}(X)$ sequentiell erzeugt wird und das Abbruch Kriterium des Algorithmus, der Grad von $R0$ ausgewertet wird. Ferner ist eine Einrichtung 12 vorhanden, die den Grad T des Fehlerort-Polynoms $T_s(X)$ ermittelt für die nachfolgende Chien Suche.

50 Der Funktionsblock 6 ermittelt in Abhängigkeit von den anliegenden Syndromen $S(X)$ und Auslöschorten $L(X)$ zugehörige Fehlerort-Polynome $T_s(X)$, Fehlerwert-Polynome $R_s(X)$ sowie ein Steuersignal über den Grad von T des Fehlerort-Polynoms $T_s(X)$. An Ausgängen des Funktionsblockes 6 sind somit abnehmbar: Signale des Fehlerort-Polynoms $T_s(X)$, die über eine 17×8 Bit parallele Datenbusleitung $Tout$ zu einem Funktionsblock 13 gelangen, Signale des Fehlerwert-Polynoms $R_s(X)$, die über eine 16×8 Bit parallele Datenbusleitung $Rout$ zu dem Funktionsblock 13 weitergereicht werden, und das Steuersignal über den Grad von T des Fehlerort-Polynoms $T_s(X)$, das über eine 5 Bit Leitung $deg[T]$ dem Funktionsblock 13 zugeführt wird. Einzelheiten zu dem Funktionsblock 6 werden nachfolgend in Zusammenhang mit den Fig. 6 bis 10 erläutert.

60 In dem Funktionsblock 13 werden die Null-Stellen der Polynome $T_s(X)$ und $R_s(X)$ bestimmt, so daß mit erhaltenen Fehlerwerten Y_k und Fehlerorten X_k eine Korrektur fehlerhafter Datenworte in dem Funktionsblock 14 durchgeführt werden kann. Der Block 13 wird auch von den von der Verzögerungseinrichtung 2 verzögerten Datenworten $del. D_{in}$ und dem verzögerten Fehlersignal $del. E_{Fin}$ angesteuert.

Der Aufbau und die Wirkungsweise der Funktionsblöcke 13 und 14 ist aus der DE 41 05 860 A1 bekannt.

65 Die Steuerung der vier Funktionsblöcke 3, 6, 13 und 14 sowie der Verzögerungseinrichtung 2 erfolgt durch eine Steuereinrichtung 15. An einem Ausgang des Funktionsblockes 14 ist auf einem Datenbus 16 ein Datenstrom mit 8 Bit breiten Datenworten D_{out} abnehmbar von denen im Rahmen der Korrekturfähigkeit des RS-Decoders erkannte fehlerhafte Datenworte korrigiert sind. Andere Datenworte, die die Korrekturfähigkeit des RS-Decoders überschreiten, werden durch ein auf einer Leitung 17 parallel übertragenes Fehlermerkersignal E_{Fout} markiert.

Der in Verbindung mit der Fig. 2 als Registerbank 7 bezeichnete Block ist in der Fig. 3 ausführlicher dargestellt. Der Block umfaßt im wesentlichen eine 16×8 -Bit-parallele Datentransfereinrichtung zur Erzeugung des Fehlerwertpolynoms $R_s(X) = (Q_{s-i}(X)R_{s-1}(X) + R_{s-1}(X))$.

Aus Gründen der Übersichtlichkeit ist nur eine einzige Ebene der 16×8 -Bit-parallelen Ebenen der vorliegenden Datentransfereinrichtung dargestellt.

In der Fig. 3 ist mit 18 ein Galois-Feld-(GF)-Multiplizierer bezeichnet. Ein Eingang dieses GF-Multiplizierers ist mit dem Ausgang eines 2/1-Multiplexers 19 und anderer Eingang mit dem Ausgang eines weiteren 2/1-Multiplexers 20 verbunden. Der Ausgang des GF-Multiplizierers 18 liegt an dem Eingang eines UND-Gatters 21, welches Ausgangsdaten des GF-Multiplizierers 18 in Abhängigkeit eines Steuersignals $EnFb[i]$ an seinem anderen Eingang zu einem Eingang eines EXOR-Gatters 22 weiterleitet. Der andere Eingang des EXOR-Gatters 22 ist über eine von einem Steuersignal $EnIn[i]$ beaufschlagtes UND-Gatter 23 mit einem 2/1-Multiplexer 24 verbunden. An einem Ausgang des EXOR-Gatters 22 abnehmbare 8 Bit breite Datenworte gelangen zu dem Eingang eines Registerstapels 25, welches die Daten speichert und um eine Taktperiode verzögert wieder ausgibt.

Die perspektivisch angedeuteten 16 Ebenen des Registerstapels 25 sollen darauf hinweisen, daß die gesamte Registertransferstruktur inclusive der Multiplexer 19, 24 und 28, des Multiplizierers 18, der EXOR- sowie AND-Gatter (21, 22, 23) sechzehn mal parallel vorhanden ist. Daher sind an einem Ausgang der vorderen Ebene des Registerstapels 25 die Daten $RO[15]$ abnehmbar, an einem Ausgang der zweiten Ebene des dazu parallelen Registerstapels 25^{II} die Daten $RO[14]$ und an einem Ausgang der 16. Ebene beispielsweise die Daten $RO[0]$. In der in der Fig. 3 dargestellten Ebene der erfindungsgemäßen Registertransfereinrichtung sind die Ausgangsdaten $RO[i]$ des Registerstapels 25 auf den Eingang 0 des 2/1-Multiplexers 24 zurückgekoppelt. An dem anderen Eingang 1 des Multiplexers 24 sind die vom Funktionsblock 3 ausgegebenen Syndrome S_i aufgeschaltet. Der 2/1-Multiplexer 24 wird durch ein Schaltsignal Insel gesteuert.

Der in eine Eingangsleitung des GF-Multiplizierers 18 eingefügte 2/1-Multiplexer 20 ist eingangsseitig mit dem Ausgang eines Konstantenspeichers 26 sowie mit dem Ausgang eines Registers 27 zur Speicherung eines Koeffizienten des Zwischenwertpolynoms $Q_{s-i}(X)$ verbunden. Das Register 27 ist Teil der in der Fig. 5 gezeigten Steuereinrichtung 11 der Registertransfers. Die Ableitung der in dem Register 27 gespeicherten Koeffizienten des Zwischenwertpolynoms $Q_{s-i}(X)$ wird in Zusammenhang mit der Beschreibung der Steuereinrichtung 11 in der Fig. 5 erläutert.

Der in die andere Eingangsleitung des GF-Multiplizierers 18 eingefügte 2/1-Multiplizierer 19 ist eingangsseitig mit dem Ausgang $RO[i-1]$ der zweiten Ebene des Registerstapels 25 sowie mit dem Ausgang eines 9/1-Multiplizierers 28 verbunden. Der 2/1-Multiplizierer 19 ist durch ein Schaltsignal R-sel und der 2/1-Multiplizierer 20 durch ein Schaltsignal H-sel steuerbar. Beide Schaltsignale erzeugt die in der Fig. 5 dargestellte Steuereinrichtung 11.

Den neun Eingängen des 9/1-Multiplexers 28 in der Ebene i, der durch ein Schaltsignal $d[3 \dots 0]$ der Steuereinrichtung 11 steuerbar ist, sind Daten von Ausgängen $R[i]$ bis $R[i-1]$ eines in der Fig. 4 gezeigten Registerstapels 29 aufgeschaltet.

Die Schaltsignale Insel, R-sel, H-sel und $d[3 \dots 0]$ gelten für alle 16 Byte der Registertransfereinrichtung gemeinsam. Die Steuersignale $EnIn[i]$ und $EnFb[i]$ werden dagegen für jede Registerebene des Registerstapels 25 in der Steuereinrichtung getrennt erzeugt.

Die Fig. 8 zeigt eine Wahrheitstabelle über das Zusammenwirken der vorgenannten Schalt- und Steuersignale für den Betrieb des Registerstapels 25. In dieser und weiteren Wahrheitstabellen bedeutet X einen beliebigen logischen Pegel, L eine logische "0", H eine logische "1" und die Ziffern 1 bis 7 die Verbindung eines bestimmten Eingangs mit dem Ausgang eines zugehörigen Multiplexers.

Gemäß der in der ersten Zeile der Wahrheitstabelle angegebenen Verknüpfung können die Schaltsignale Insel, $d[3 \dots 0]$, R-sel und H-sel einen beliebigen Pegel X und die Steuersignale $EnIn[i]$ und $EnFb[i]$ einen L-Pegel aufweisen, um in einer Initialisierungsphase, d. h. zu Beginn einer Datenblockübertragung, den Registerstapel 25 mit dem Hex-Wert 00 zu laden. In der zweiten Zeile der in der Fig. 8 dargestellten Wahrheitstabelle sind die logischen Pegel der Schalt- und Steuersignale angegeben, um den Registerstapel 25 mit den in dem Funktionsblock 3 ermittelten Syndromen $S[i]$ zu laden. In der Wahrheitstabelle folgen sodann Verknüpfungen, um die in dem Registerstapel 25 gespeicherten Daten zu halten, um die Daten in dem Registerstapel 29 zu vertauschen und um eine Erzeugung des Fehlerwertpolynoms $R_s(X)$ zu veranlassen.

Soll bei dem Registerstapel 29 der Ausgangswert $RO[i]$ ($i = 0 \dots 15$) mit dem Ausgangswert $R(i-j)$ verknüpft werden und ist der Index negativ, so ist an den 2/1-Multiplexer 24 der Hex-Wert 00 anzulegen. Der Index "j" bezeichnet dabei eine Laufvariable (1..8) für den 9/1-Multiplexer 28.

Der in der Fig. 4 dargestellte Registerstapel 29 besteht ebenso wie der Registerstapel 25 aus insgesamt 16 Ebenen zur Speicherung von 16×8 Bit breiten Datenworten für die Dauer einer Taktperiode. Der Registerstapel 29 wird über ein UND-Gatter 30 selektiv mit Daten angesteuert, die an vier Eingängen eines 4/1-Multiplexers 31 liegen. Bei der hintersten Ebene des Registerstapels 29 $R[0]$ liegen am Eingang des Multiplexers beispielsweise die Ausgangsdaten $R[1]$ des Registerstapels 29, die Auslöschorddaten $L[0]$, die Ausgangsdaten $RO[0]$ des Registerstapels 25 und die zurückgekoppelten Ausgangsdaten $R[0]$ des Registerstapels 29. Die Steuerung des 4/1-Multiplexers 31 erfolgt durch ein in der Steuereinrichtung 11 abgeleitetes Schaltsignal $Rinsel[1 \dots 0]$. Die an einem Ausgang des 4/1-Multiplexers 31 abnehmbaren Daten werden in Abhängigkeit eines dem UND-Gatter 30 zugeführten Steuersignals $EnRin[i]$ zu dem Eingang des Registerstapels 29 weitergeleitet.

Das Registerstapel 29 wählt für alle Ebenen gemeinsam einen der vier Eingänge des 4/1-Multiplexers 31 an. Darüberhinaus besteht die Möglichkeit, jedes der 16 Bytes des Registerstapels 29 getrennt anzusprechen; dies geschieht durch eine entsprechende Steuerung des in die Eingangsleitung des Registerstapels 29 eingefügten UND-Gatters 30 durch ein Schaltsignal $EnRin[i]$, damit das entsprechende Byte auf den Hex-Wert 00 gesetzt

werden kann. Ferner kann das Registerstapel 29 mit Auslöschorten $L[i]$ geladen werden; es kann weiterhin veranlaßt werden, einen gespeicherten Wert zu halten, seinen Speicherinhalt mit den des Registerstapels 25 zu tauschen und den Wert $R[i]$ um eine Stelle zu schieben. Bei einem Schieben wird der Wert $R[0]$ weggeworfen und der Datenwert $R[15]$ von Registerstapel 29 zu "0" gesetzt.

- 5 Eine entsprechende Wahrheitstabelle zur Steuerung der obengenannten Registertransfers des Registerstapels 29 ist in der Fig. 9 dargestellt.

Der Funktionsablauf der in den Fig. 3 und 4 gezeigten Blöcke ist folgender:

- Der an dem Ausgang des 2/1-Multiplexers 20 vorliegende Koeffizient des Zwischenwertpolynom $Q_{s-i}(X)$, dessen Ableitung in Verbindung mit der Fig. 5 noch näher erläutert wird, wird in dem GF-Multiplizierer 18 mit dem vom Ausgang des 2/1-Multiplexers 19 erhaltenen, Fehlerwertpolynom $R_{s-i}(X)$ multipliziert. An dem
10 Ausgang des GF-Multiplizierers 18 steht somit der Term $(Q_{s-i}(X))R_{s-i}(X)$ an. Zu diesem Term wird mit Hilfe des EXOR-Gatters 22 das Fehlerwertpolynom $R_{s-2}(X)$ addiert, das von dem Ausgang des Registerstapels 25 über den 2/1-Multiplexer 24 und dem UND-Gatter 23 zu dem Eingang des EXOR-Gatters 22 gelangt. An dem Ausgang (Klemme 32) des EXOR-Gatters 22 steht das durch Anwendung eines Euclid-Algorithmus erzeugte
15 Fehlerwert-Polynom $R_s(X) = (Q_{s-i}(X))R_{s-i}(X) + R_{s-2}(X)$ zur Weiterverarbeitung in dem Funktionsblock 13 des RS-Decoders bereit.

- Das zur Erzeugung des Fehlerwertpolynoms $R_s(X)$ benötigte Zwischenwertpolynom $Q_{s-i}(X) = R_{s-2}(X)/R_{s-i-1}(X)$ wird in der Steuereinrichtung 11 (Fig. 5) durch eine Division ermittelt. Die zur Division erforderliche Anordnung besteht aus einem 3/1-Multiplexer 33, einem Galois-Feld-Inverter 34, einem Multiplizierer 35, einem
20 2/1-Multiplexer 36, einem 5/1-Multiplexer 37, einem Register 27, einem 17/1-Multiplexer 39, einem weiteren Multiplizierer 40, einem 3/1-Multiplexer 41 sowie einem Register 42.

- Bei dieser Anordnung ist einem Eingang des 17/1-Multiplexers 39 das 16×8 Bit breite Signal RO des Fehlerwertpolynoms $R_{s-2}(X)$, das am Ausgang des Registerstapels 25 (Fig. 3) abnehmbar ist, aufgeschaltet. Von dem Signal RO wird ein Teilsignal $RO[1]$ zu einem Eingang des 3/1-Multiplexers 33 geleitet. Weitere Eingänge des
25 3/1-Multiplexers 33 sind mit einem Signal der Auslöschorte $L[0]$ und dem Ausgang des Registers 42 beaufschlagt. Das Ausgangssignal des 3/1-Multiplexers 33 wird über den Galois-Feld-Inverter 34 zu dem Eingang des Multiplizierers 35 geleitet, so daß eine Division, entsprechend der Multiplikation mit der Inversen, berechnet werden kann. Ein anderer Eingang des Multiplixers 35 ist mit dem Ausgang des 2/1-Multiplexers 36 verbunden, dessen
30 einer Eingang auf einem logischen H-Pegel liegt und dessen anderem Eingang ein Signal über die verwendete Codewortlänge CWlen zugeführt ist. Ein Ausgangssignal h-quot des Multiplizierers 35 gelangt zu Eingängen 0, 1 und 3 des 5/1-Multiplexers 37 sowie zu dem Eingang 0 des 3/1-Multiplexers 41. Der Ausgang des 5/1-Multiplexers 37 ist mit einem Eingang des Registers 27 verbunden, dessen Ausgangssignal $Q_{s-i}(X)$ auf der Leitung Q auf den Eingang 4 des 4/1-Multiplexers 37 zurückgekoppelt ist.

- Dem anderen Eingang des 17/1-Multiplexers 39 ist ein Signal mit dem Hex-Wert 00 zugeführt. Das Ausgangssignal leading_term[RO] des 17/1-Multiplexers 39 wird zu einem Eingang des Multiplizierers 40 und zu dem
35 Eingang 2 des 3/1-Multiplexers 41 weitergereicht. Daten am Ausgang des 3/1-Multiplexers 41 werden in das Register 42 geschrieben, dessen Ausgangssignal h0 zu einem anderen Eingang des Multiplizierers 40, zu einem Eingang 1 des 3/1-Multiplexers 41 und zu einem Eingang 2 des 3/1-Multiplexers 33 gelangt. Die Steuerung des 17/1-Multiplexers 39 erfolgt durch ein Schaltsignal, das in einer Stufe 43 zur Ableitung des Grads von RO erzeugt wird.

- 40 Ferner enthält die Steuereinrichtung 11 eine Anordnung zur Beendigung der Rechenoperationen, d. h. zum Abbruch des Euclid-Algorithmus. Diese Anordnung ist mit einer Stufe 43 zur Ableitung des Grads von RO, einem Multiplexer 44, einem Binäraddierer 45, einem 4/1-Multiplexer 46, einem 5-Bit-Register 47, einem 3/1-Multiplexer 48, der Registereinrichtung 12, einer NICHT-Stufe 49 und einem UND-Gatter 50 ausgestattet.
45 Dabei liegt ein Eingang 3 des Multiplexers 4 an einem Ausgang der Stufe 43 und ein Eingang 2 an einer Leitung P, auf welcher die Anzahl der verwendeten Parity-Worte übertragen wird. Das von dem Multiplexer 44 abgegebene Ausgangssignal gelangt zu einem Eingang des Binär-Addierers 45. An einem Eingang "Carry in" (Eingangsübertrag) des Binär-Addierers 45 liegt der logische Pegel "1". Der weiterer Eingang des Binär-Addierers 45 ist mit dem Ausgang der NICHT-Stufe 49 verbunden. Ein am Ausgang des Binär-Addierers 45 abnehmbares Signal wird zu dem Eingang 2, 3 des 4/1-Multiplexers 46 weitergeleitet, dessen Eingang 1 den logischen
50 Pegel "1" führt und dessen Eingang 0 an dem Ausgang des 5-Bit-Registers 47 angeschlossen ist. Der Ausgang des 4/1-Multiplexers 46 liegt an dem Eingang des 5-Bit-Registers 47. Vom Ausgang des 5-Bit-Registers 47 gelangt das Signal sodann über das durch ein Steuersignal dmask beeinflusste UND-Gatter 50 zur Ausgangsleitung d[3 ... 0], die die 9/1 Multiplexer (28,60) von TO und RO steuert.

- 55 Das Ausgangssignal der Stufe 43 wird auch dem Eingang 2 des 3/1-Multiplexers 48 zugeführt. Der Eingang 1 des 3/1-Multiplexers 48 erhält ein Signal von der 5-Bit-Registereinrichtung 12 und der Eingang 0 des 3/1-Multiplexers 48 ein Signal mit dem Hex-Wert 00. Der Ausgang des 3/1-Multiplexers 48 führt auf den Eingang der 5-Bit-Registereinrichtung 12, die ein Signal $\text{deg}[RO]$ ausgibt und dieses Signal auf Eingänge der NICHT-Stufe 49 und des 3/1-Multiplexers 48 zurückführt. Somit wird die binäre Subtraktion des Grades von RO ermöglicht. Ist der Grad von RO kleiner als eine im Algorithmus vorgegebene Schwelle, so ist die Berechnung abzubrechen.
60 Das Ergebnis steht in den Registerstapeln 29 (Fig. 4) und 61 (Fig. 7) für die weitere Verarbeitung in der Funktionseinrichtung 13 zur Verfügung.

- Die Abkürzung $\text{deg}(RO)$ bezeichnet den Grad des Polynoms RO des Registerstapels 25, also die Position des höchsten Koeffizienten ungleich Null. Der Grenzwert in dgr ist laut Algorithmus $(P + E - 2) \text{ DIV } 2$. Dann ist
65 Euclid terminiert.

Das 5-Bit-Register 47 läßt sich mit Werten (P-dgr), $(\text{deg}(RO)\text{-dgr})$, 01 und d laden. Der Wert des Registers 12 ergibt sich aus dem Wert $\text{deg}(RO)$, dgr oder 00. Das Register 42 kann geladen werden mit dem Hex-Wert 00, Werten h0 am Ausgang des Registers 42, einem Wert INV(h0) und dem höchstwertigen Koeffizienten von RO.

Dabei ist $INV = X^{-1} = 1/x$ das reziproke Element für die Division im Galois Feld. Das Register 27 ist ladbar mit dem Wert (h0*führender_Term(RO)). Mit den vorgenannten Registern und einem weiteren (nicht dargestellten) Zustandszähler läßt sich ein Mikroprogramm-Steuerwerk aufbauen, welches die Multiplexer der Registerstapel in der Steuereinrichtung 11 steuert und den Euclid-Algorithmus entsprechend anwendet; es kann insbesondere festgestellt werden, ob der Wert "dgr" kleiner als ein Grenzwert ist und ob der Wert "d" negativ ist, wobei das höchstwertige Bit abgefragt wird.

Außerdem enthält die Steuereinrichtung 11 eine logische Verknüpfungseinrichtung 51, an welcher anhand der auf Ausgangsleitungen dgr, d[3...0] und Q zugeführten Signalen und des aktuellen Schritts im Euclid-Algorithmus die in den Fig. 3, 4, 6 und 7 benötigten Steuer- und Schaltsignale für die einzelnen Multiplexer und Addier-Gatter erzeugt und abnehmbar sind. Die von der Steuereinrichtung 11 gelieferten Signale sind: Insel, Rinsel[1...0], R_sel, H_sel, Set1, T_sel, EnT, EnIn[16...0], EnFb[16...0] und EnRin[15...0].

Die Fig. 6 zeigt ein Blockschaltbild einer Registertransfereinrichtung zur Erzeugung des Fehlerortpolynoms $T_s(X)$. Die Struktur des Blockschaltbildes ähnelt der zur Erzeugung des Fehlerwertpolynoms $R_s(X)$ gemäß der Fig. 3. Gleiche Elemente sind auch hier mit gleichen Bezugszeichen versehen. In der Fig. 6 bezeichnet 52 einen GF-Multiplizierer, dessen einer Eingang mit dem Ausgang eines 2/1-Multiplexers 53 und dessen anderer Eingang mit dem Ausgang eines weiteren 2/1-Multiplexers 54 verbunden ist. Der Ausgang des GF-Multiplizierers 53 liegt an dem Eingang eines UND-Gatters 55, welches Ausgangsdaten des GF-Multiplizierers 52 in Abhängigkeit eines das UND-Gatter 55 steuernden Steuersignals EnFb[i] einem Eingang eines EXOR-Gatters 56 zuleitet. Der andere Eingang des EXOR-Gatters 56 ist über ein von dem Steuersignal EnIn[i] beaufschlagtes UND-Gatter 57 verbunden, dem Ausgangsdaten T0[i] eines Registerstapels 58 zugeführt sind. Dieser Registerstapel 58 weist 17 Ebenen für jeweils 8 Bit breite Datenworte auf. Auch hier ist aus Gründen der Übersichtlichkeit nur eine Ebene dargestellt. In der ersten Ebene bzw. Stufe T0[0] des Registerstapels 58 ist in die Verbindungsleitung zwischen dem Ausgang des UND-Gatters 57 und einem Eingang des EXOR-Gatters 56 ein ODER-Gatter 59 eingefügt, das den Signalstrom durch ein Steuersignal Set1 beeinflussen kann.

Der in eine Eingangsleitung des GF-Multiplizierers 52 eingefügte 2/1-Multiplexer 54 ist eingangsseitig mit dem Ausgang des Konstantenspeichers 26 sowie mit dem Ausgang des bereits erwähnten Registers 27 verbunden, das einen Koeffizienten des Zwischenwertpolynoms $Q_{s-1}(X)$ enthält. Der in der anderen Eingangsleitung des GF-Multiplizierers 52 eingefügte 2/1-Multiplexer 53 ist eingangsseitig mit dem am Ausgang T0[i-1] der zweiten Ebene des Registerstapels 58 sowie dem am Ausgang eines 9/1-Multiplexers 60 verbunden. Der 2/1-Multiplexer 53 ist durch ein Schaltsignal T_sel und der 2/1-Multiplexer 54 durch das Schaltsignal H_sel steuerbar. Beide Schaltsignale werden in der Steuereinrichtung 11 (Fig. 5) erzeugt.

Den neun Eingängen des 9/1-Multiplexers 60, der durch das Schaltsignal d3...0 der Steuereinrichtung 11 steuerbar ist, sind Daten zugeführt, die an Ausgängen T[i] bis T[i-8] eines in der Fig. 7 dargestellten Registerstapels 61 abgreifbar sind. Die Steuersignale EnIn[i] und EnFb[i] ermöglichen eine separate Ansteuerung der einzelnen Registerebenen des Registerstapels 58. Die Schaltsignale Set1, T_sel, H_sel und d gelten für alle 17 Bytes des Registerstapels 58 gemeinsam. Ebenso gilt Tinsel und EnT für alle 17 Ebenen des Registerstapels 61.

Die Fig. 9 zeigt eine Wahrheitstabelle über das Zusammenwirken dieser einzelnen Schalt- und Steuersignale zum Betrieb des Registerstapels 58. Zum Verständnis dieser Wahrheitstabelle seien nachfolgend die ersten beiden Zeilen betrachtet. In der ersten Zeile der Wahrheitstabelle findet zunächst eine Initialisierung des Registerstapels 58 statt, bei welcher in die 17 Ebenen des Registerstapels 58 der Hex_Wert 00 geschrieben wird. Zur Einleitung des Initialisierungsvorgang müssen die Steuersignale Set1, EnIn[i] und EnFb[i] einen logischen L-Pegel aufweisen, während die Schaltsignale d, T_sel und H_sel einen beliebigen Pegel X annehmen dürfen. Die zweite Zeile der Wahrheitstabelle nimmt Bezug auf die Besonderheit der Initialisierung der ersten Ebene des Registerstapels 58. Dabei wird der Hex-Wert 01 in die erste Ebene des Registerstapels 58 geschrieben, während die übrigen Ebenen den Hex-Wert 00 annehmen. Die folgenden Zeilen der Wahrheitstabelle enthalten Kombinationen, die das Halten und Vertauschen von Registerinhalten und die steuernden Maßnahmen zum Erhalt des Fehlerortpolynoms $T_s(X)$ betreffen.

Der in der Fig. 7 dargestellte Registerstapel 61 besteht ebenso wie der Registerstapel 58 aus 17 parallelgeschalteten Speicherebenen in einer Wortbreite von jeweils 8 Bit. Auch der Registerstapel 61 speichert die zugeführten Datenworte für die Dauer einer Taktperiode. Das Registerstapel 61 wird über ein UND-Gatter 62 selektiv von Daten angesteuert, die an zwei Eingängen eines 2/1-Multiplexers 63 liegen. Von den Eingängen der 2/1-Multiplexer 63 ist der Eingang 1 mit einem Signal T0[i] am Ausgang des Registerstapels 58 und der Eingang Null mit dem Signal T[i] am Ausgang des Registerstapels 61 beaufschlagt. Die Steuerung des 2/1-Multiplexers 63 erfolgt durch ein in der Steuereinrichtung 11 abgeleitetes Schaltsignal Tinsel. Von dem 2/1-Multiplexer 63 ausgegebenen Daten gelangen über das UND-Gatter 62, das durch ein Steuersignal EnT freigegeben wird, zu dem Eingang des Registerstapels 61.

Eine zur Steuerung des Registertransfers von Registerstapel 61 zugeordnete Wahrheitstabelle zeigt die Fig. 11. Je nach logischem Pegel der Steuersignale Tinsel und EnT wird der Inhalt des Registerstapels 61 beeinflusst, d. h. in einer Initialisierungsphase wird der Registerstapel 61 mit dem Hex-Wert 00 geladen, die in dem Registerstapel gespeicherte Dateninformation wird gehalten bzw. mit der im Register 58 gespeicherten Dateninformation getauscht.

Die in den Fig. 6 und 7 dargestellten Registertransfereinrichtungen weisen folgende Funktion auf:

Das an dem Ausgang des 2/1-Multiplexers 54 abnehmbare Zwischenwertpolynom $Q_{s-1}(X)$ wird in dem GF-Multiplizierer 52 mit dem vom Ausgang des 2/1-Multiplexers 53 abnehmbaren Fehlerortpolynom $T_{s-1}(X)$ multipliziert. Das Ausgangssignal des GF-Multiplizierers 52 genügt dem Term $(Q_{s-1}(X))T_{s-1}(X)$. Diesem Term wird in dem EXOR-Gatter 56 das Fehlerortpolynom $T_{s-2}(X)$ zuaddiert, so daß am Ausgang (Klemme 64) des EXOR-Gatters 56 das durch Anwendung eines Euclid-Algorithmus erzeugte Fehlerortpolynom $T_s(X) = Q_{s-1}(X)T_{s-1}(X) + T_{s-2}(X)$ zur Weiterverarbeitung in den Funktionsblock 13 des RS-Decoders zur Verfügung steht.

Patentansprüche

1. Reed-Solomon-Decoder mit

- einer Einrichtung (3) zur Ermittlung von Syndromen ($S(X)$) und Auslöschorten ($L(X)$) in einem zu decodierenden Datensignal,
- einer Einrichtung (6) zur Erzeugung von Fehlerort- und Fehlerwert-Polynomen anhand ermittelter Syndrome ($S(X)$) und Auslöschorte ($L(X)$) nach einem Euclid-Algorithmus, bei welchem

$$T_s(X) = (Q_{s-1}(X))T_{s-1}(X) + T_{s-2}(X)$$

$$R_s(X) = (Q_{s-1}(X)R_{s-1}(X) + R_{s-2}(X) \text{ und}$$

$$Q_{s-1}(X) = R_{s-2}(X)/R_{s-1}(X) \text{ ist,}$$

wobei $T_s(X)$ ein Fehlerortpolynom, $R_s(X)$ ein Fehlerwertpolynom und $Q_{s-1}(X)$ ein Zwischenwertpolynom ist, und welche Einrichtung (6) einen ersten und zweiten Registerstapel (25, 29) zur Speicherung von zwei zuvor erzeugten Fehlerwertpolynomen ($R_{s-1}(X)$, $R_{s-2}(X)$) sowie einen dritten und vierten Registerstapel (58, 61) zur Speicherung von zwei zuvor erzeugten Fehlerortpolynomen ($T_{s-1}(X)$, $T_{s-2}(X)$) aufweist,

- einer Einrichtung (13) zur Ermittlung von Fehlerorten X_k und Fehlerwerten Y_k durch eine Nullstellensuche nach Chien in den erzeugten Fehlerort- und Fehlerwertpolynomen $T_s(X)$ und $R_s(X)$ und
- einer Einrichtung (14) zur Korrektur von Datenworten in dem zu decodierenden Datensignal anhand der ermittelten Fehlerorte X_k und Fehlerwerte Y_k ,

dadurch gekennzeichnet, daß zur Erzeugung des Fehlerwertpolynoms $R_s(X)$ vorgesehen sind:

ein erster Galois-Feld-Multiplizierer (18), dessen einem Eingang Daten vom Ausgang eines ersten Registers (27) mit Koeffizienten eines Zwischenwertpolynoms $Q_{s-1}(X)$ und dessen anderem Eingang Daten ($R[i]$) vom Ausgang des zweiten Registerstapels (29) abgreifbaren Fehlerwertpolynoms $R_{s-1}(X)$ aufgeschaltet ist, wobei einem Eingang des zweiten Registerstapels (29) Ausgangsdaten des ersten Registerstapels (25) zugeleitet sind, und

eine erste Einrichtung (22) zur Addition der an einem Ausgang des ersten Galois-Feld-Multiplizierers (18) vorliegenden Daten zu Daten an einem Ausgang des ersten Registerstapels (25), daß zur Erzeugung des Fehlerortpolynoms $T_s(X)$ vorgesehen sind:

ein zweiter Galois-Feld-Multiplizierer (52), dessen einem Eingang Daten das an dem Ausgang des ersten Registers (27) abnehmbaren Koeffizienten des Zwischenwertpolynoms $Q_{s-1}(X)$ und dessen anderem Eingang Daten $T[i]$ des an einem Ausgang des vierten Registerstapels (61) abgreifbare Fehlerortpolynom $T_s(X)$ aufgeschaltet ist, wobei einem Eingang des vierten Registerstapels (61) Ausgangsdaten des dritten Registerstapels (58) zugeleitet sind, und

eine zweite Einrichtung (56) zur Addition der an einem Ausgang des zweiten Galois-Feld-Multiplizierers (52) vorliegenden Daten zu Daten an einem Ausgang des dritten Registerstapels (58), und daß zur Erzeugung des Zwischenwertpolynoms $Q_{s-1}(X)$ vorgesehen ist:

eine Einrichtung (33 bis 40) zur Division der Ausgangsdaten des zweiten Registerstapels (29) durch die Ausgangsdaten des ersten Registerstapels (25), welche einen durch Division erhaltenen Quotienten als Zwischenwertpolynom ($Q_{s-1}(X)$) dem ersten Register (27) zur Speicherung zuführt.

2. Reed-Solomon-Decoder nach Anspruch 1, dadurch gekennzeichnet, daß die erste Einrichtung (22) zur Addition ein achtfaches EXOR-Gatter (22) mit jeweils zwei Eingängen ist, bei welchem ein Eingang mit dem Ausgang des ersten Galois-Feld-Multiplizierers (18) und ein anderer Eingang mit dem Ausgang des ersten Registerstapels (25) verbunden ist und bei welchem der Ausgang an einem Eingang des ersten Registerstapels (25) liegt.

3. Reed-Solomon-Decoder nach Anspruch 1 und 2, gekennzeichnet durch eine erste und zweite Einrichtung (21, 23) zur UND-Verknüpfung von den den Eingängen des ersten EXOR-Gatters (22) zugeführten Daten mit jeweils einem ersten und zweiten Steuersignal ($EnIn[i]$, $nFb[i]$), welche einem Eingang der ersten und zweiten Einrichtung (21, 23) zur UND-Verknüpfung zugeführt sind, um den Datenfluß im Sinn einer Unterbrechung zu steuern.

4. Reed-Solomon-Decoder nach Anspruch 1 bis 3, gekennzeichnet durch einen ersten Multiplexer (24) mit zwei Eingängen und einem Ausgang, bei welchem gesteuert durch ein erstes Schaltsignal (Insel) entweder die am Ausgang des ersten Registerstapels (25) abnehmbaren Daten oder die ermittelten Syndrome (S_i) zu dem einen Eingang des ersten EXOR-Gatters (22) leitbar sind.

5. Reed-Solomon-Decoder nach Anspruch 1, gekennzeichnet durch einen zweiten Multiplexer (20) mit zwei Eingängen und einem Ausgang, über welchen gesteuert durch ein zweites Schaltsignal (H_sel) entweder die am Ausgang des ersten Registers (27) abnehmbaren Daten des Zwischenwertpolynoms $Q_{s-1}(X)$ oder der Wert einer Konstanten ($\alpha^0 = 1$) zu dem einen Eingang des ersten Galois-Feld-Multiplizierers (18) leitbar ist,

einen dritten Multiplexer (19) mit zwei Eingängen und einem Ausgang, über welchem gesteuert durch ein drittes Schaltsignal (R_sel), entweder die am Ausgang des ersten Registerstapels (25) abnehmbaren Daten ($R0[i-1]$) oder bestimmte Daten ($R[i]$, $R[i-1]$, ..., $R[i-8]$) des Fehlerwertpolynoms $R_s(X)$ vom Ausgang des zweiten Registerstapels (R) zu dem anderen Eingang des ersten Galois-Feld Multiplizierers (18) leitbar sind, und

einen durch ein fünftes Schaltsignal ($d[3 \dots 0]$) gesteuerten vierten Multiplexer (28) mit neun Eingängen und einem Ausgang, dessen Eingänge verschiedene an Ausgängen des zweiten Registerstapels (29) liegenden Datenworte ($R[i]$, $R[i-1]$, ..., $R[i-8]$) des Fehlerwertpolynoms $R_s(X)$ zugeführt sind und dessen Ausgang mit einem der beiden Eingänge des dritten Multiplexers (19) verbunden ist.

6. Reed-Solomon-Decoder nach Anspruch 1, gekennzeichnet durch einen durch ein sechstes Schaltsignal ($R_{\text{insel}}[1 \dots 0]$) gesteuerten fünften Multiplexer (31) mit vier Eingängen und einem Ausgang, bei welchem einem Eingang ein erstes bestimmtes Datenwort (R_{i+1}) des am Ausgang des zweiten Registerstapels (29) abnehmbaren Fehlerwertpolynoms $R_s(X)$ zugeführt ist, bei welchem einem zweiten Eingang ein bestimmtes Datenwort ($L[i]$ der Auslöschungsorte) zugeführt ist, bei welchem einem dritten Eingang ein bestimmtes an dem Ausgang des ersten Registerstapels (25) liegendes Datenwort ($RO[i]$) zugeführt ist und bei welchem einem vierten Eingang ein zweites bestimmtes Datenwort ($R[i]$) des zweiten Registerstapels (29) abnehmbaren Fehlerwertpolynoms $R_s(X)$ zugeführt ist, und eine dritte Einrichtung (30) zur UND-Verknüpfung von am Ausgang des fünften Multiplexers (31) abnehmbaren Daten mit einem dritten Steuersignal ($EnR_{\text{in}}[i]$) und Weiterleitung der durch die UND-Verknüpfung gewonnenen Daten zu einem Eingang des zweiten Registerstapels (29). 5
7. Reed-Solomon-Decoder nach Anspruch 1, dadurch gekennzeichnet, daß die zweite Einrichtung (56) zur Addition ein zweites achtfaches EXOR-Gatter (56) mit jeweils zwei Eingängen (56) ist, bei welchem ein Eingang mit dem Ausgang des zweiten Galois-Feld-Multiplizierers (52) und ein anderer Eingang mit dem Ausgang des dritten Registerstapels (58) verbunden ist und bei welchem der Ausgang an einem Eingang des dritten Registerstapels (61) angeschlossen ist. 15
8. Reed-Solomon-Decoder nach Anspruch 1 und 7, gekennzeichnet durch eine vierte und fünfte Einrichtung (55, 57) zur UND-Verknüpfung von den den Eingängen den zweiten EXOR-Gatters (56) zugeführten Daten mit jeweils dem ersten und zweiten Steuersignal ($EnI_{\text{in}}[i]$, $EnF_{\text{b}}[i]$), welche einem Eingang der vierten und fünften Einrichtung (55, 57) zur UND-Verknüpfung aufgeschaltet sind, um den Datenfluß im Sinn einer Unterbrechung zu steuern. 20
9. Reed-Solomon-Decoder nach Anspruch 7 und 8, dadurch gekennzeichnet, daß zwischen dem Ausgang der vierten Einrichtung (57) zur UND-Verknüpfung und dem einen Eingang des zweiten EXOR-Gatters (56) eine Einrichtung zur ODER-Verknüpfung (59) eingefügt ist, um in einer Stufe $TO(0)$ des dritten Registerstapels (58) eine logische "1" in den Datenstrom einzufügen. 25
10. Reed-Solomon-Decoder nach Anspruch 7 bis 9, gekennzeichnet durch einen sechsten Multiplexer (54) mit zwei Eingängen und einem Ausgang, über welchem gesteuert durch das zweite Schaltsignal (H_{sel}) entweder die am Ausgang des ersten Registers (27) abnehmbaren Koeffizienten des Zwischenwertpolynoms $Q_{s-i}(X)$ oder der Wert einer Konstanten ($\alpha^0 = 1$) zu dem einem Eingang des zweiten Galois-Feld-Multiplizierers (52) leitbar ist, 30
- einen siebten Multiplexer (53) mit zwei Eingängen und einem Ausgang, über welchem gesteuert durch ein viertes Schaltsignal (T_{sel}) entweder die am Ausgang des dritten Registerstapels (58) abnehmbaren Daten ($TO[i-1]$) oder bestimmte Daten ($T[i]$, $T[i-1]$ $T[i-8]$) des Fehlerortpolynoms $T_s(X)$ vom Ausgang des vierten Registerstapels (61) zu dem anderen Eingang des zweiten Galois-Feld Multiplizierers (52) leitbar sind, und einen durch das fünfte Schaltsignal ($d[3 \dots 0]$) gesteuerten achten Multiplexer (60) mit neun Eingängen und einem Ausgang, dessen Eingänge verschiedene, an Ausgängen des vierten Registerstapels (61) liegende Datenworte ($T[i]$, $T[i-1]$, \dots , $T[i-8]$) des Fehlerortpolynoms $T_s(x)$ zugeführt sind und an dessen Ausgang mit einem der beiden Eingänge des siebten Multiplexers (53) verbunden ist. 35
11. Reed-Solomon-Decoder nach Anspruch 1, gekennzeichnet durch einen durch ein siebentes Schaltsignal (T_{insel}) gesteuerten neunten Multiplexer (63) mit zwei Eingängen und einem Ausgang, bei welchem einem ersten Eingang ein bestimmtes, an dem Ausgang des dritten Registerstapels (58) liegende Datenwort ($TO[i]$) zugeführt ist, und bei welchem einem zweiten Eingang ein bestimmtes, an dem Ausgang des vierten Registerstapels (61) liegendes Datenwort ($T[i]$) zugeführt ist, und eine sechste Einrichtung (62) zur UND-Verknüpfung von am Ausgang des neunten Multiplexers (63) abnehmbaren Daten mit einem vierten Steuersignal (EnT) und zur Weiterleitung der UND-verknüpften Daten zu einem Eingang des vierten Registerstapels (61). 40
12. Reed-Solomon-Decoder nach Anspruch 1, gekennzeichnet durch eine Steuereinrichtung (11) zur Ableitung der Schalt- und Steuersignale in Abhängigkeit des Grades einer Division, die aus dem Divisor $R_{s-2}(X)$ und dem Divident $R_{s-1}(X)$ des Zwischenwertpolynoms $Q_{s-1}(X)$ gebildet ist. 45
13. Reed-Solomon-Decoder nach Anspruch 1 und 12, dadurch gekennzeichnet, daß die Einrichtung (33 bis 40) zur Division einen ersten Multiplizierer (35) aufweist, dessen einem Eingang über einen Galois-Feld-Inverter (34) Auslöschorte ($L(X)$) und dessen anderen Eingang ein Wert über die Codewortlänge (CW_{len}) des zu decodierenden Datensignals zugeführt sind und an dessen Ausgang als Quotient das Zwischenwertpolynom $Q_{s-i}(X)$ abnehmbar ist. 50
14. Reed-Solomon-Decoder nach Anspruch 1, dadurch gekennzeichnet, daß zur Erzeugung des fünften Schaltsignals ($d[3 \dots 0]$) vorgesehen sind: 55
- ein zweites Register (12) zur Speicherung von Daten über den Grad des in dem ersten Registerstapel (25) gespeicherten Fehlerwertpolynoms $R_{s-2}(X)$,
- ein Subtrahierer (49, 52), welchem alternativ ein Wert (P) über die Anzahl von verwendeten Parities in Datenworten des zu decodierenden Datensignals oder Daten über den Grad des in dem ersten Registerstapel (25) gespeicherten Fehlerwertpolynoms $R_{s-2}(X)$ sowie Daten an einem Ausgang des zweiten Registers (12) zugeführt sind, und 60
- einem dritten Register (47) zur Speicherung des von dem Subtrahierer (49, 52) abgegebenen Schaltsignals ($d[3 \dots 0]$). 65

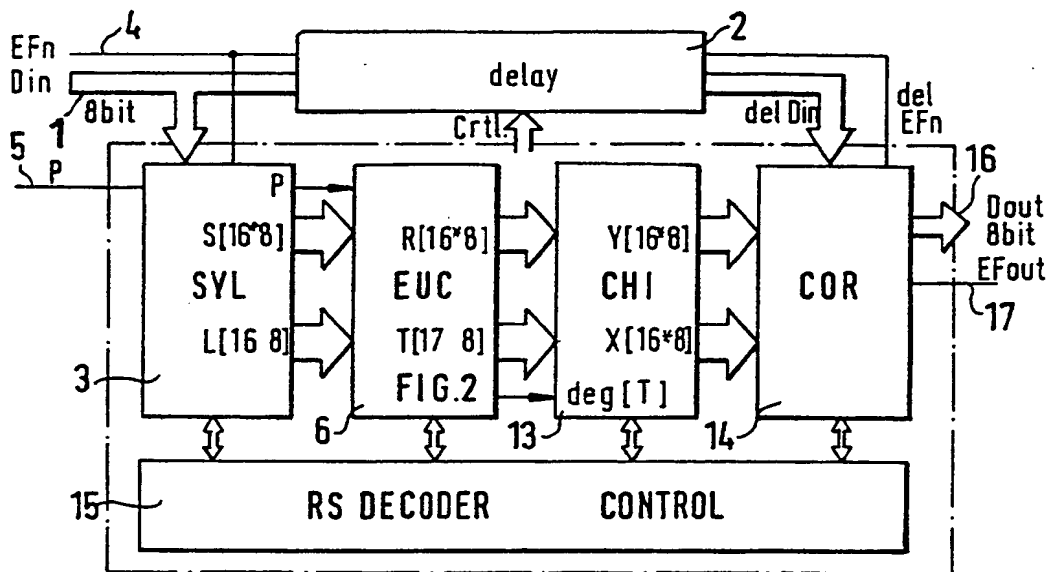


Fig. 1

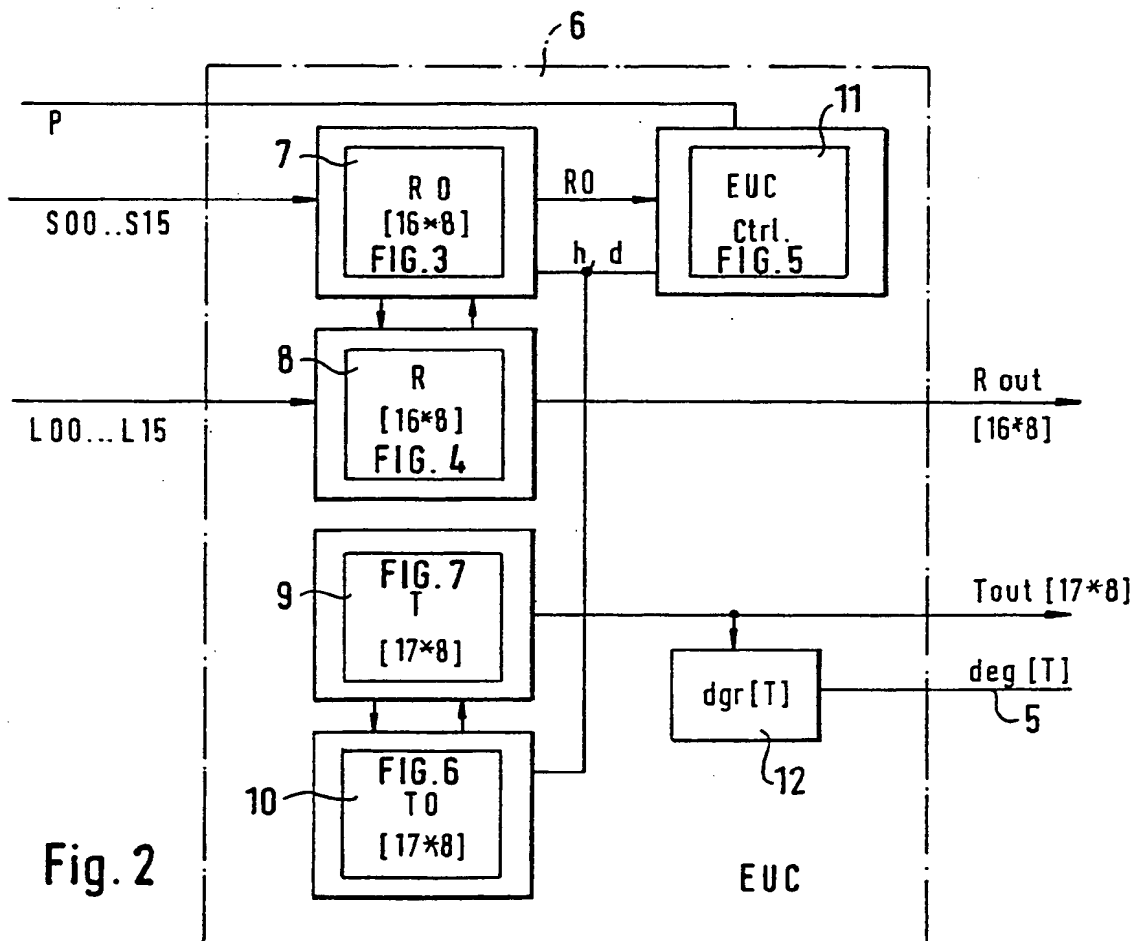
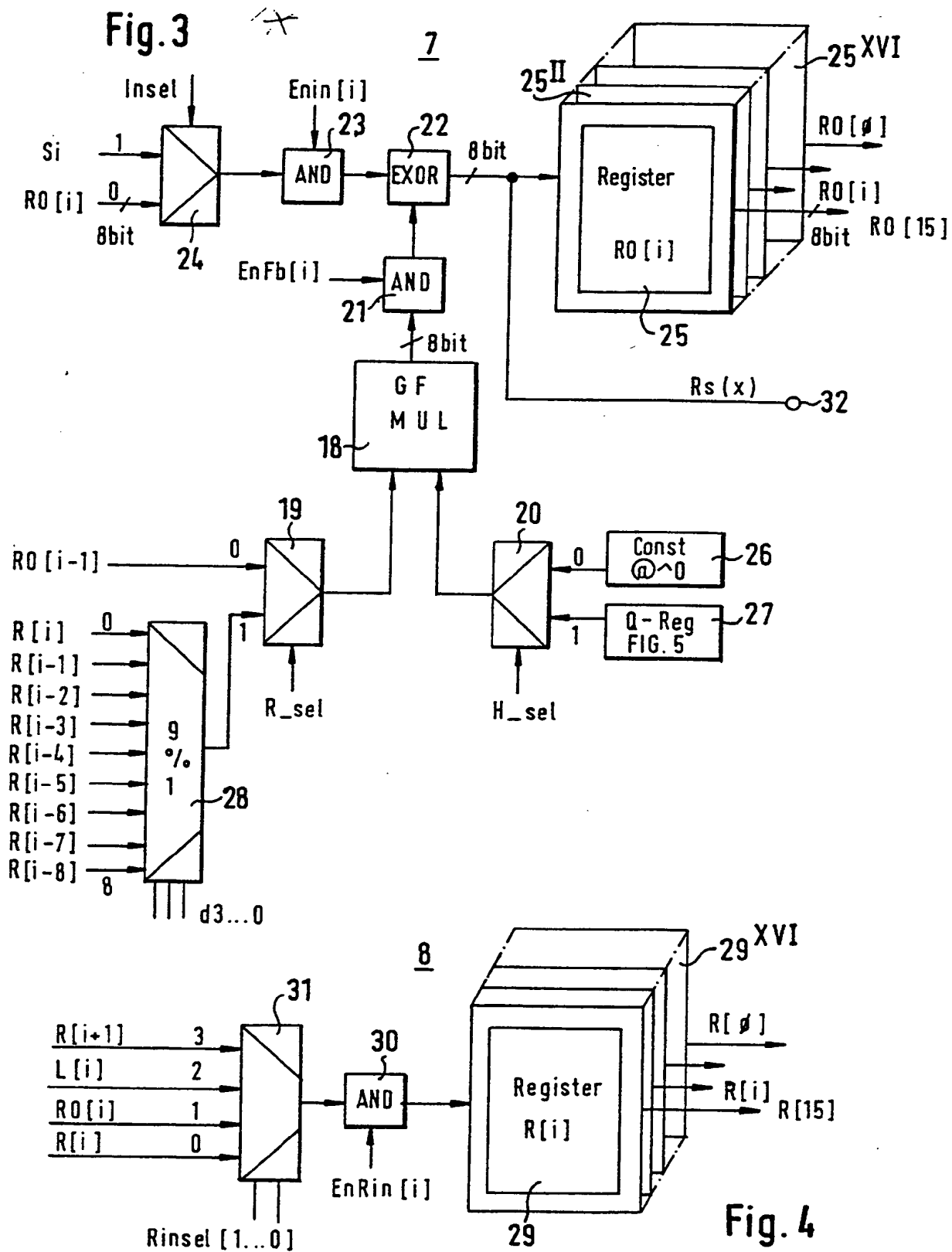


Fig. 2



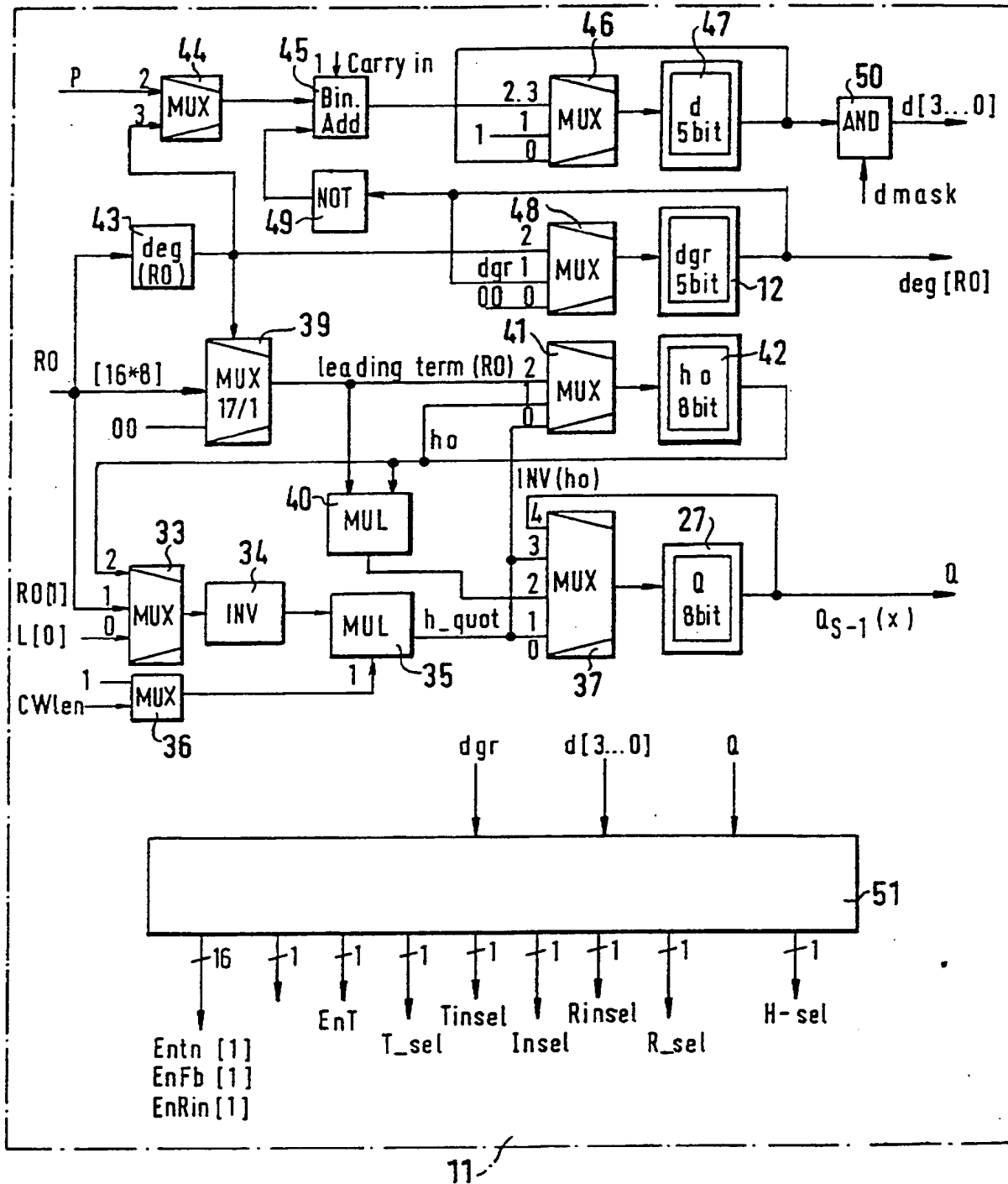


Fig. 5

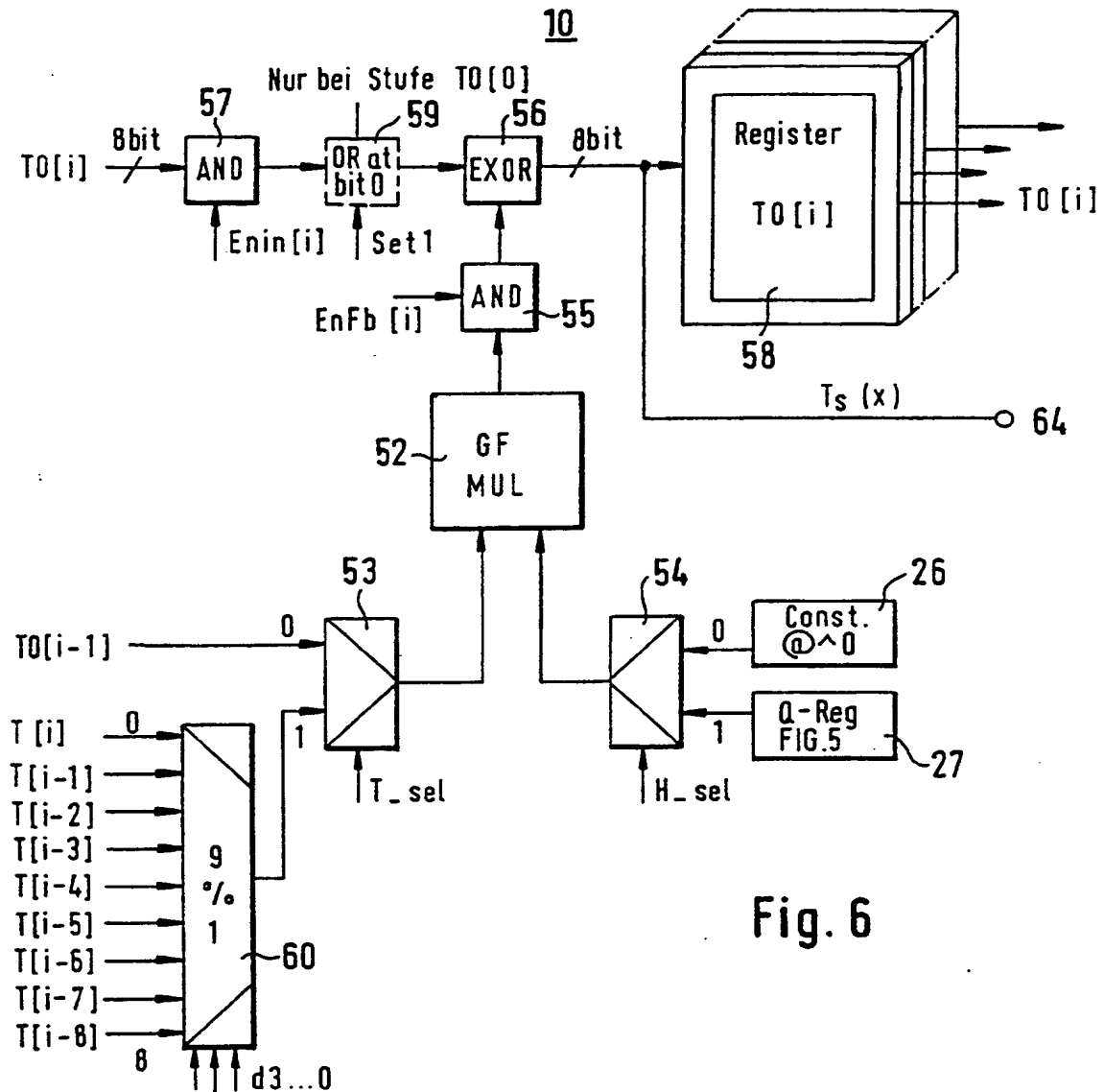


Fig. 6

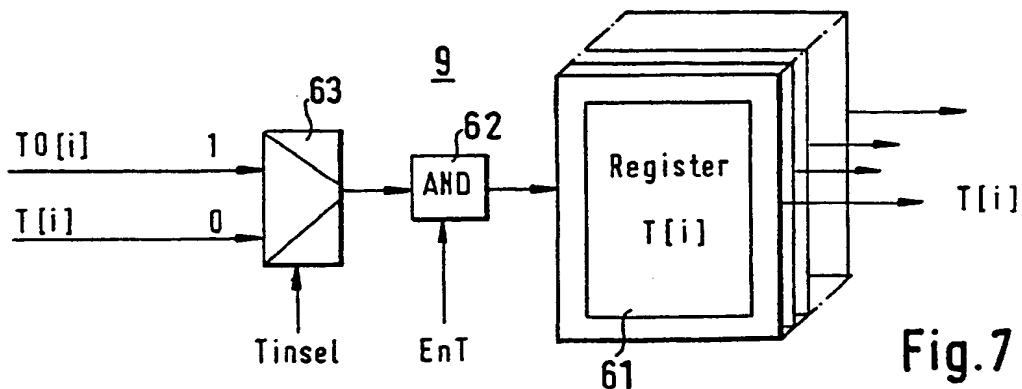


Fig. 7

Register Transfers von Registerstapel 25

Insel	EnIn	EnFb	d	Rsel	Hsel	R0[i]	
X	L	L	X	X	X	\$00	(CLEAR)
H	H	L	X	X	X	S[i]	(LOAD)
L	H	L	X	X	X	R0[i]	(HOLD)
X	L	H	0	H	L	R[i]	(SWAP)
L	H	H	0	H	H	R0[i] + R[i]	* Q
L	H	H	1	H	H	R0[i] + R[i-1]	* Q
L	H	H	2	H	H	R0[i] + R[i-2]	* Q
L	H	H	3	H	H	R0[i] + R[i-3]	* Q
L	H	H	4	H	H	R0[i] + R[i-4]	* Q
L	H	H	5	H	H	R0[i] + R[i-5]	* Q
L	H	H	6	H	H	R0[i] + R[i-6]	* Q
L	H	H	7	H	H	R0[i] + R[i-7]	* Q
L	H	H	8	H	H	R0[i] + R[i-8]	* Q
L	H	H	0	H	L	R0[i] + R[i]	
L	H	H	1	H	L	R0[i] + R[i-1]	
L	H	H	2	H	L	R0[i] + R[i-2]	
L	H	H	3	H	L	R0[i] + R[i-3]	
L	H	H	4	H	L	R0[i] + R[i-4]	
L	H	H	5	H	L	R0[i] + R[i-5]	
L	H	H	6	H	L	R0[i] + R[i-6]	
L	H	H	7	H	L	R0[i] + R[i-7]	
L	H	H	8	H	L	R0[i] + R[i-8]	

Fig. 8

Register Transfers von Registerstapel 58

Set1	EnIn	EnFb	d	Tsel	Hsel	T0[i]
L	L	L	X	X	X	T0[16..0]=\$00 (Clear)
H	L	L	X	X	X	T0[16..1]=\$00; T0[0]=\$01
L	H	L	X	X	X	T0[i] (Hold)
L	H	H	0	H	L	T[i] (SWAP)
L	H	H	0	H	H	T0[i] + T[i] * Q
L	H	H	1	H	H	T0[i] + T[i-1] * Q
L	H	H	2	H	H	T0[i] + T[i-2] * Q
L	H	H	3	H	H	T0[i] + T[i-3] * Q
L	H	H	4	H	H	T0[i] + T[i-4] * Q
L	H	H	5	H	H	T0[i] + T[i-5] * Q
L	H	H	6	H	H	T0[i] + T[i-6] * Q
L	H	H	7	H	H	T0[i] + T[i-7] * Q
L	H	H	8	H	H	T0[i] + T[i-8] * Q
L	H	H	0	H	L	T0[i] + T[i]
L	H	H	1	H	L	T0[i] + T[i-1]
L	H	H	2	H	L	T0[i] + T[i-2]
L	H	H	3	H	L	T0[i] + T[i-3]
L	H	H	4	H	L	T0[i] + T[i-4]
L	H	H	5	H	L	T0[i] + T[i-5]
L	H	H	6	H	L	T0[i] + T[i-6]
L	H	H	7	H	L	T0[i] + T[i-7]
L	H	H	8	H	L	T0[i] + T[i-8]

Fig. 9

Register Transfers von Registerstapel 29

EnRin[i]	Rinsel	R[i]	

L	X	\$00	(CLEAR)
H	0	R[i]	(HOLD)
H	1	RO[i]	(SWAP)
H	2	L[i]	(LOAD)
H	3	R[i+1]	(SHIFT)

Fig. 10

Register Transfers von Registerstapel 61

Tinsel	EnT	R[i]	

X	L	\$00	(CLEAR)
L	H	T[i]	(HOLD)
H	H	TO[i]	(SWAP)

Fig. 11